REMARKS

This Amendment is responsive to the Final Office Action dated July 14, 2006. Applicant has amended claims 1-3, 5, 6, 8, 17, 18 and 20-24. Claims 1-8, 10-18 and 20-25 are pending.

Claim Rejection Under 35 U.S.C. § 103

In the Final Office Action, the Examiner rejected claims 1-8, 10-18 and 20-25 under 35 U.S.C. § 103(a) as being unpatentable over US 6,996,631 to Aiken Jr. et al. (Aiken) in view of US 6,128,657 to Okanoya et al. (Okanoya). Applicant respectfully traverses this rejection to the extent the rejection may be considered applicable to the claims as amended. The applied references fail to disclose or suggest the inventions defined by Applicant's claims, and provide no teaching that would have suggested the desirability of modification to arrive at the claimed invention.

Applicant has amended a number of the claims to clarify the distinctions between the requirements of the claims and the teachings of Aiken and Okanoya. Applicant respectfully submits that Aiken and Okanoya fail to disclose or suggest the requirements of Applicant's claims as amended for all of the reasons set forth in the after-final Response dated October 16, 2006. Rather than merely reiterate those reasons for allowability, Applicant provides the following remarks to address the arguments made by the Examiner in the Advisory Action dated December 4, 2006.

Most of the Claims Do Not Specify the Usage of Sockets

Applicant respectfully disagrees with the assertion that the claims as previously presented did not recite the usage of sockets. The independent claims as previously presented did recite sockets. Nevertheless, Applicant has amended claim 1 to make clear that the plurality of server TCP connections from an intermediate networking device correspond to a plurality of sockets on the physical servers. For example, amended claim 1 recites that the intermediate device comprises an HTTP multiplexor/demultiplexor configured to monitor response parameters specific to individual ones of the plurality of server TCP connections from the intermediate networking device to the plurality of corresponding sockets on the physical server devices.

Amended claim 1 further recites that the intermediate networking device further comprises a plurality of agents. Claim 1 requires that, upon receiving an HTTP request, the respective agent identifies a single one of plurality of physical servers specified as a destination within the HTTP request. According to claim 1, the agent then identifies at least two server TCP connections that couple the intermediate computing device to a plurality of different sockets on the same physical device specified as the destination within the HTTP request. Thus, claim 1 literally requires the agent identifies two or more TCP connections that couple the intermediate device to *different sockets* on the *same* physical device that is specified as the *destination* for the request.

In addition, claim 1 requires that the agent selects one of the identified server TCP connections that couple the intermediate computing device to the *different* sockets on the *same* physical device specified as the *destination* within the HTTP request. Claim 1 requires that the selection is based on the monitored response parameters specific to the different sockets on the physical server device specified as the destination within the HTTP request. Thus, claim 1 literally requires that the selection is based on response parameters specific to *different socket on the same physical server device*, where that physical server device is specified as the *destination for the HTTP request.*

Finally claim 1 requires that the agent routes the HTTP request over the selected one of the TCP connections that couple the intermediate computing device to the different sockets on the same physical device specified as the destination within the HTTP request for communication to the physical server device as a multiplexed HTTP request. Thus, claim 1 literally requires that the agent route the multiplexed HTTP request to the selected one of the TCP connections that couple the intermediate computing device to the different sockets on the same physical device specified as the destination within the HTTP request.

For the reasons discussed in Applicant's previous response, Aiken and Okanoya fail to disclose or suggest these requirements of independent claim 1. Instead, they disclose load distribution (i.e., load balancing) at the physical server or stack level. In other words, Aiken and Okanoya teach load balancing across different physical devices based on the attributes of those physical devices. Okanoya and Aiken do not suggest monitoring response parameters for TCP connections to different sockets on the same physical server. Nor do the references teach or

suggest identifying two or more TCP connections that couple the intermediate device to *different sockets* on the *same* physical device that is specified as the *destination* for the request, as required by claim 1. Furthermore, the load balancing / multiplexing techniques of the cited references fail to provide any teaching or suggestion of selecting one of the identified server TCP connections that couple the intermediate computing device to the *different* sockets on the *same* physical device specified as the *destination* within the HTTP request, as required by claim 1. The device-level load balancing described by the references provide no teaching or suggestion of selecting an individual socket based on response parameters specific to *different socket on the same physical server device*, where that physical server device is specified as the *destination for the HTTP request*, as required by claim 1.

Independent claims 3, 6, 17, 18, 22 and 24 have been similarly amended to clarify the distinctions between the requirements of those claims and the teachings of Aiken and Okanoya.

Not Limited to a Single Physical Server / Multiplexing Within a Server / Preamble

In the Advisory Action, the Examiner asserted that "load balancing works as multiplexing" and that the claims "do not limit the system to a single physical server (not counting gateways)." The Examiner also asserted that the claims do not recite any limitation stating that the "multiplexing occurs within a server, let alone bars multiplexing to different servers." The Examiner added that "the only requirement is that there is a t least one physical server that will eventually receive the connection."

Applicant submits that the clarifying amendments overcome any of these concerns. Specifically, claim 1 is now expressly drawn to an intermediate computing device in which an agent determines the physical device that is the destination for the HTTP request, and then identify at least two at least two server TCP connections that couple the intermediate computing device to a plurality of different sockets on that same physical device specified as the destination within the HTTP request. Claim 1 further requires that the agent selects one of the identified server TCP connections that couple the intermediate computing device to the *different* sockets on the *same* physical device specified as the *destination* within the HTTP request.

For at least these reasons the claims are not drawn to a system that only multiplexes or load balances to different servers based on considerations of those servers. Instead, as set forth in

detail above, the claims specifically require that the intermediate device include an agent that identifies at least two different TCP connections to different sockets to the same physical device, where that physical device is specified as the ultimate destination for the HTTP request. Claim 1 further requires that the agent then select one of the two or more TCP connections to that physical device based on <u>monitored</u> response parameters specific to the <u>different sockets on the physical server device</u> specified as the destination within the HTTP request. Consequently, the Examiner's concern that claims cover systems, such as Okanova, that only contemplate multiplexing or load balancing to different servers based on considerations of those servers should be moot.

Furthermore, the Examiner is correct that Applicant's claims do not bar the existence of another server, or distribution of clients requests to another server. The Examiner is also correct that Applicant's claims do not require multiplexing within a server. However, Applicant submits that these concerns are not relevant to the analysis of Applicant's claims as clarified. It is what the claims do require, and the fact that what is required by the claims is not taught or suggested by the Aiken and Okanoya that is relevant to the analysis of Applicant's claims.

Intended Use

In the Advisory Action, the Examiner noted that intended use is not a patentable limitation. Applicant is confused by this statement. None of Applicant's claims recites an intended use. Instead, Applicant's claims recite structural and functional limitations that must be given patentable weight.

Distribution of Requests at the Connection/Socket Level

The Examiner asserted that the claims as previous presented did not required distribution of requests at the connection/socket level. Applicant respectfully disagrees.

Moreover, as discussed above, Applicant has amended the independent claims to clarify the distinctions between their requirements and the teachings of Aiken and Okanoya. For the reasons discussed above and in the previous response, Aiken and Okanoya fail to disclose or suggest the requirements of Applicant's amended claims.

Agents

The Examiner argued that the agents within servers taught by Okanoya (Fig. 2; nos. 13, 23 and 33) are each associated with a client TCP connection, as required by independent claims 1 and 18 as previously presented. Applicant respectfully disagrees. The claims as previously presented clearly required that the agents are part of the intermediate networking device. Directly contrary to this clear requirement of Applicant's claims as previously presented, the Okanoya agents are located within in the servers of the Okanoya system, instead of within the intermediate device of the Okanoya system.

Further, Applicant has amended claims 1 and 18 to clarify that each of the plurality of agents is assigned to a different one of the plurality of client TCP connections from the client to the intermediate networking device. With respect to Fig. 2, Okanoya does not suggest that the agents 13, 23 and 33 are assigned to respective ones of connections between device 100 and clients 63, 64.

Moreover, the agents in Okanoya do not identify at least two server TCP connections that couple device 100 to a plurality of different sockets on the same physical device 10, 20 or 30 specified within an HTTP request, as required by independent claims 1 and 18. Moreover, the Okanoya agents do not select one of the identified server TCP connections based on monitored response parameters specific to the different sockets on that same physical device, as further required by independent claims 1 and 18 as amended. To the contrary, as discussed above, the Okanoya agents merely report the functionality of their respective server *as a whole*. There is no suggestion that an Okanoya agent individually considers a plurality of sockets of or connections to its server.

Neither Aiken nor Okanoya discloses or suggests these numerous requirements with respect to agents. Thus, the Examiner has failed to establish a prima facie case of obviousness of at least claims 1 and 18.

Aiken's Usage of Ports

Aiken describes a cluster of data processing systems 20, 24, 28 and 36 having a plurality of different protocol stacks 22, 26, 30, 34 and 38 that utilize a single IP address. See, e.g., FIG. 4. Specifically, each data processing systems 20, 24, 28, and 36 are operating system images

executing on one or more computer systems. Col. 8, ll. 49-55. Thus, each of data processing systems 20, 24, 28 and 36 may be a separate physical computing device or the logical equivalent of a physical computing device. A routing protocol stack associates a Virtual IP Address (VIPA) and a port with each of the other communication protocol stacks in the cluster and routes communications to the appropriate protocol stack based on the VIPA and the particular port. Col. 8, ll. 29-40. Each data processing system (20, 24, 28, 36), utilizes its own protocol stack and the entire processing system communications from the routing protocol stack for the data processing systems VIPA and port.

In this manner, Aiken teaches nothing more than load balancing across a plurality of data processing systems, whether physically separate computing systems or the logical equivalent of physically separate computing systems. Aiken does not teach or suggest an intermediate device in which an agent identifies at least two different TCP connections from the intermediate device to different sockets to the same physical device, where that physical device is specified as the ultimate destination for the HTTP request, as required by claim 1. Furthermore, Aiken does not teach or suggest an agent that selects one of the two or more TCP connections to that physical device based on monitored response parameters specific to the different sockets on the physical server device specified as the destination within the HTTP request, as required by claim 1.

Even assuming that the data processing systems of Aiken were implemented as logically separate data processing systems incorporated in the same physical computing system, Aiken would fail to teach or suggest many of the features recited by Applicant's claims. As one example, Aiken's multiplexing within a server across logically separate data processing systems fails to teach or suggest identifying at least two different TCP connections from an intermediate device to different sockets to the same physical device, where that physical device is specified as the ultimate destination for the HTTP request, as required by claim 1.

The data processing systems in Aiken are not specified as destinations within HTTP requests. To the contrary, data processing systems hidden behind a single IP address and are instead assigned their own Virtual IP address. Thus, the data processing systems 22, 24, 28 and 36 in Aiken would be hidden from clients 46 and could not be individually specified as destinations by client 46. See, FIG. 4. For this reason, Aiken does not teach or suggest an intermediate device capable of identifying at least two different TCP connections from an

RECEIVED
CENTRAL FAX CENTER
JAN 16 2007

intermediate device to different sockets to the same destination, as specified by an HTTP request, and then selecting one of the server TCP connections to that same destination based on monitored response parameters for different sockets to that specified destination, as required by claim 1.

Moreover, even if data processing systems 22, 24, 28 and 36 were somehow considered separate destinations specified within an HTTP request, then Aiken does not teach or suggest identifying and then selecting one of at least two different TCP connections from an intermediate device to different sockets to the *same destination*. To the contrary, Aiken only describes load balancing across the separate data processing systems. In Aiken, for a given data processing systems, all communications for that system are delivered to its protocol stack based on that systems VIPA and particular port assigned to that system.

For at least these reasons, and those presented in Applicant's previous Response, the Examiner has failed to establish a prima facie case for non-patentability of Applicant's claims 1-8, 10-18 and 20-25 under 35 U.S.C. § 103(a). Withdrawal of this rejection is requested.

CONCLUSION

All claims in this application are in condition for allowance. Applicant respectfully requests reconsideration and prompt allowance of all pending claims. Please charge any additional fees or credit any overpayment to deposit account number 50-1778. The Examiner is invited to telephone the below-signed attorney to discuss this application.

Date: By:

January 16, 2007

SHUMAKER & SIEFFERT, P.A.
8425 Seasons Parkway, Suite 105
St. Paul, Minnesota 55125
Telephone: 651.735.1100
Facsimile: 651.735.1102

Name: Kent J. Sieffert
Reg. No.: 41,312